

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 63-314647

(43)Date of publication of application : 22.12.1988

(51)Int.Cl.

G06F 9/46  
G06F 9/46

(21)Application number : 62-150471

(71)Applicant : NEC CORP

(22)Date of filing : 17.06.1987

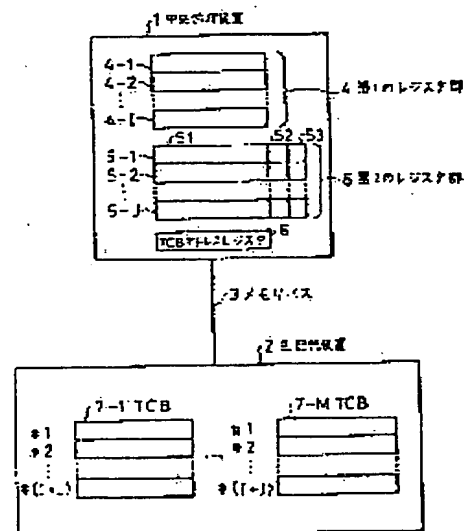
(72)Inventor : OTAKA MASAYUKI

## (54) SAVING/RESTORING SYSTEM FOR TASK RUN ENVIRONMENT

## (57)Abstract:

**PURPOSE:** To reduce the overhead caused when tasks are switched without decreasing the number of environment information holding registers, by dividing plural registers into 1st and 2nd groups and specifying the saving and restoring of the 2nd register group.

**CONSTITUTION:** Plural registers holding the environment information are divided into 1st and 2nd register groups 4 and 5. The registers of the group 4 are saved and stored as conventional; while only the registers that undergone replacement during the task run are saved among those registers of the group 5. Then the registers of the group 5 are restored at a time point when the reference occurs in a run mode and under the condition that said reference is the first one and no replacement is carried out before the first reference. Thus it is possible to reduce the overhead when tasks are switched and to switch these tasks at a high speed.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭63-314647

⑬ Int.Cl.<sup>4</sup>

G 06 F 9/46

識別記号

3 1 3  
3 4 0

庁内整理番号

A-7056-5B  
B-7056-5B

⑭ 公開 昭和63年(1988)12月22日

審査請求 未請求 発明の数 1 (全5頁)

⑮ 発明の名称 タスク走行環境退避復元方式

⑯ 特 願 昭62-150471

⑰ 出 願 昭62(1987)6月17日

⑱ 発 明 者 大 阪 正 之 東京都港区芝5丁目33番1号 日本電気株式会社内

⑲ 出 願 人 日本電気株式会社 東京都港区芝5丁目33番1号

⑳ 代 理 人 弁理士 境 廣 巳

#### 明 細 書

##### 1. 発明の名称

タスク走行環境退避復元方式

##### 2. 特許請求の範囲

タスク切替時、中央処理装置内に設けられている複数のレジスタに保持されている中断タスクの環境情報を退避させ、再開タスクの環境情報を前記複数のレジスタに復元するタスク走行環境退避復元方式に於いて、

前記複数のレジスタを第1のレジスタ群と第2のレジスタ群とに分割し、

タスク切替時、前記第1のレジスタ群に保持されている中断タスクの環境情報の全て及び前記第2のレジスタ群に保持されている中断タスクの環境情報の内の該中断タスクが起動されてから中断されるまでの間に更新されたレジスタに保持されている環境情報のみを退避させ、再開タスクの環境情報の内の前記第1のレジスタ群の環境情報を前記第1のレジスタ群に復元し、

前記再開タスクの起動後、前記第2のレジスタ

群に含まれるレジスタが参照された場合、前記再開タスクが起動されてから今回の参照が行なわれるまでの間に該レジスタが更新も参照もされていないことを条件として、該参照されたレジスタに前記再開タスクの環境情報を復元することを特徴とするタスク走行環境退避復元方式。

##### 3. 発明の詳細な説明

(産業上の利用分野)

本発明はマルチタスク処理を行なうコンピュータシステムに於けるタスク走行環境退避復元方式に関する。

(従来の技術)

マルチタスク処理を行なうコンピュータシステムに於いては、タスク切替時、タスク走行環境を退避、復元させる必要がある。

第5図は従来のタスク走行環境退避復元方式を説明するためのブロック図である。

中央処理装置10と主記憶装置11とはメモリバス12により接続され、中央処理装置10は走行中のタスクの環境情報を保持するN個のレジスタ13-1

## 特開昭63-314647(2)

～13-Nと、レジスタ13-1～13-Nに保持されている現在進行中のタスクの環境情報を退避させるタスクコントロールブロック（以下TCBと称す）のアドレスがセットされるTCBアドレスレジスタ14とを含み、主記憶装置11はM個のTCB15-1～15-Mを含んでいる。尚、TCB15-1～15-Mは同時に走行するタスクの数だけ、各タスク対応に設けられているものである。

中央処理装置10は走行中のタスクが中断すると、次に走行可能なタスクを再開させるために、レジスタ13-1～13-Nに保持されている上記中断したタスクについてのN個の環境情報をTCBアドレスレジスタ14が指し示す主記憶装置11上のTCBに退避させる。今、例えば、TCBアドレスレジスタ14がTCB15-1を指し示しているとする、レジスタ13-1～13-NからTCB15-1へのメモリ書込みがN回つまりレジスタ13-1～13-Nの個数分発生することになる。

次に中央処理装置10はタスクの走行を再開させるために、上記再開させるタスク対応のTCBの

内容をレジスタ13-1～13-Nに復元する。今、例えば、再開させるタスク対応のTCBがTCB15-Mであるとする、TCB15-Mからレジスタ13-1～13-Nへのメモリ読出しがN回発生することになる。以上のように、1回のタスク切替で中央処理装置10と主記憶装置11との間でN回ずつの読出し、書込みが生じる。

（発明が解決しようとする問題点）

上述した従来例は、タスク切替時に中央処理装置10に設けられている環境情報保持用のレジスタ13-1～13-Nの数と同じ回数だけ中央処理装置10と主記憶装置11との間で読出し、書込みを行わなければならない、環境情報保持用のレジスタの数が多くなるとオーバーヘッドが大きくなってしまいう問題がある。また、環境情報保持用のレジスタ13-1～13-Nの数を少なくすれば、オーバーヘッドを小さくすることはできるが、一般的に中央処理装置10の処理能力は環境情報保持用のレジスタ13-1～13-Nの数が多いほど高いものとなるので、オーバーヘッドを小さくするためにレジ

スタ数を減少させたのでは、中央処理装置10の処理能力が低下してしまう問題がある。

本発明は前述の如き問題点を解決したものであり、その目的は環境情報保持用のレジスタの数を減少させることなく、タスク切替時のオーバーヘッドを小さくできるようにすることにある。

（問題点を解決するための手段）

本発明は前述の如き問題点を解決するため、

タスク切替時、中央処理装置内に設けられている複数のレジスタに保持されている中断タスクの環境情報を退避させ、再開タスクの環境情報を前記複数のレジスタに復元するタスク走行環境退避復元方式に於いて、

前記複数のレジスタを第1のレジスタ群と第2のレジスタ群とに分割し、

タスク切替時、前記第1のレジスタ群に保持されている中断タスクの環境情報の全て及び前記第2のレジスタ群に保持されている中断タスクの環境情報の内の該中断タスクが起動されてから中断されるまでの間に更新されたレジスタに保持され

ている環境情報のみを退避させ、再開タスクの環境情報の内の前記第1のレジスタ群の環境情報を前記第1のレジスタ群に復元し、

前記再開タスクの起動後、前記第2のレジスタ群に含まれるレジスタが参照された場合、前記再開タスクが起動されてから今回の参照が行なわれるまでの間に該レジスタが更新も参照もされていないことを条件として、該参照されたレジスタに前記再開タスクの環境情報を復元する。

（作用）

第2のレジスタ群に含まれるレジスタについての退避処理は、中断タスクが起動されてから中断されるまでの間に更新されたレジスタについてのみ行なわれる。第2のレジスタ群に含まれるレジスタについての復元処理は、再開タスクの起動後、参照があった時点で、それが最初の参照であり、且つその参照以前に更新がされていない場合のみ行なわれる。

（実施例）

次に本発明の実施例について図面を参照して説

## 特開昭63-314647(3)

明する。

第1図は本発明の実施例のブロック図である。

中央処理装置1と主記憶装置2とはメモリバス3により接続されている。中央処理装置1は走行中のタスクの環境情報を保持する第1、第2のレジスタ群4、5と、現在走行中のタスクの環境情報を退避させるTCBのアドレスがセットされるTCBアドレスレジスタ6を含んでいる。第1のレジスタ群4はタスクの再開時に常に参照される環境情報(例えば、プログラムカウンタのカウント値、プログラム状態語等)がセットされる1個のレジスタ4-1~4-1から構成され、第2のレジスタ群5はその他の環境情報がセットされるJ個のレジスタ5-1~5-Jから構成され、第2のレジスタ群5を構成する各レジスタ5-1~5-Jはそれぞれ環境情報がセットされる領域51と、有効フラグがセットされる領域52と、更新フラグがセットされる領域53とを含んでいる。また、主記憶装置2は中央処理装置1上で同時に走行する各タスク対応のTCB7-1~7-Mを含

んでいる。尚、レジスタ5-1~5-Jの領域52にセットされる有効フラグはレジスタ5-1~5-Jの内容が走行中タスクの環境下で有効か否かを示すフラグであり、レジスタ5-1~5-Jの内容が更新、復元された場合にONに設定され、レジスタ5-1~5-Jの退避処理時にOFFにされる。また、レジスタ5-1~5-Jの領域53にセットされる更新フラグはレジスタ5-1~5-Jの内容がタスク再開後に更新されたか否かを示すフラグであり、レジスタ5-1~5-Jの内容が更新された場合にONとなり、レジスタ5-1~5-Jの退避処理時にOFFにされる。

第2図~第4図は中央処理装置1の処理例を示す流れ図であり、以下各図を参照して本実施例の動作を説明する。

中央処理装置1は走行中のタスクが中断すると、次に走行可能なタスクを再開させるために、先ず第2図の流れ図に示す処理を行なう。即ち、中央処理装置1は、第1のレジスタ群4の各レジスタ4-1~4-1の内容をTCBアドレスレジスタ

6が指し示すTCBに退避させ(ステップS21)、次いで、第2のレジスタ群5内のレジスタ5-1にセットされている更新フラグの状態を判定する(ステップS22、S23)。

ステップS23に於いてレジスタ5-1にセットされている更新フラグの状態がONであると判定した場合は、中央処理装置1はレジスタ5-1の内容をTCBアドレスレジスタ6が指し示すTCBに退避させた後(ステップS24)、レジスタ5-1の更新フラグ及び有効フラグを共にOFFとする(ステップS25、S26)。また、ステップS23に於いてレジスタ5-1にセットされている更新フラグの状態がOFFであると判定した場合は、中央処理装置1はレジスタ5-1の有効フラグをOFFとする(ステップS26)。

第2のレジスタ群5中の1つのレジスタに対して上述した処理を行なうと、中央処理装置1は第2のレジスタ群5に含まれる全てのレジスタ5-1~5-Jについて上述した処理を行なったか否かを判断し(ステップS27、S28)、判断結果が

NOの場合はステップS23の処理に戻り、残りのレジスタに対して上述したと同様の処理を行ない、判断結果がYESの場合は第2図の流れ図に示す処理を終了する。

第2図の流れ図に示す処理を行なうことにより、第1のレジスタ群4に含まれる各レジスタ4-1~4-1の内容及び第2のレジスタ群5に含まれるレジスタ5-1~5-Jの内の更新フラグがONとなっているレジスタ(タスク再開後に更新の行なわれたレジスタ)の内容が、TCBアドレスレジスタ6によって指し示される主記憶装置2上のTCBに退避させられることになる。つまり、更新の行なわれていないレジスタの内容は対応するTCBに保存されている内容と同一であるので、更新の行なわれたレジスタの内容のみをTCBに退避させることにより、無駄なメモリアクセスを省略するようにしている。

上述したようにして、中断タスクの環境情報を主記憶装置2上の対応するTCBに退避させると、中央処理装置1は先ず再開させるタスク対応のT

## 特開昭63-314647(4)

CBのアドレスをTCBアドレスレジスタ6に設定すると共に再開タスク対応のTCBに保持されている環境情報の内の第1のレジスタ群4対応の環境情報を第1のレジスタ群4に復元し、その後再開タスクを起動する。即ち、再開タスクの起動時には第1のレジスタ群4にのみ再開タスクの環境情報を復元し、第2のレジスタ群5には再開タスクの環境情報を復元しないものである。

再開タスクを起動した後に第2のレジスタ群5に含まれるレジスタ5-J(J=1, 2, …J)に対して更新が発生すると、中央処理装置1は第3図の流れ図に示すように、レジスタ5-Jの更新フラグ及び有効フラグを共にONとする(ステップS31, S32)。

また、再開タスクを起動した後に第2のレジスタ群5に含まれるレジスタ5-Jに対して参照が発生すると、中央処理装置1は第4図の流れ図に示すようにレジスタ5-Jの有効フラグの状態を判定する(ステップS41)。ステップS41に於いて有効フラグがOFFであると判定した場合は、中

央処理装置1はレジスタ5-Jの有効フラグをONとした後(ステップS42)、主記憶装置2上のTCBに保持されていた対応する環境情報をレジスタ5-Jに復元し(ステップS43)、ステップS41に於いて有効フラグがオンであると判定した場合は他の制御ステップの処理を行なう。

ここで、レジスタ5-Jの有効フラグはレジスタ5-Jが更新された場合及びレジスタ5-Jが更新されることなく参照された場合にONにされるものであり、中央処理装置1はレジスタ5-Jが参照された場合、レジスタ5-Jの有効フラグがOFFであることを条件としてレジスタ5-Jに環境情報を復元するようにしているものであるから、レジスタ5-Jへの環境情報の復元は、タスク再開後にレジスタ5-Jの内容が更新されておらず、且つレジスタ5-Jがタスク再開後に最初に参照された時点に於いてのみ行なわれることになる。このように、タスク再開後に更新されておらず、且つ今回の参照がタスク再開後の最初の参照であるレジスタについてののみ環境情報の復元を行なう

ことにより、無駄なメモリアクセスを省略するようにしたものである。

## (発明の効果)

以上説明したように、本発明は、環境情報を保持する複数のレジスタを第1、第2のレジスタ群に分割し、第1のレジスタ群に含まれるレジスタについての退避、復元処理は従来例と同様とし、第2のレジスタ群に含まれるレジスタについての退避処理はタスク走行中に更新が行なわれたレジスタについてのみ行ない、第2のレジスタ群に含まれるレジスタについての復元処理は走行中に参照が生じた時点で、しかもその参照が最初の参照であり、且つそれ以前に更新が行なわれていないことを条件として行なうものであるから、従来例に比較してタスク切替時のオーバーヘッドを小さなものとすることができ、タスク切替を高速化できる効果がある。また更に、メモリアストラヒックの分散化によるシステムスループットの向上を図ることができる効果もある。

## 4.図面の簡単な説明

第1図は本発明の実施例のブロック図、

第2図は退避時の処理例を示す流れ図、

第3図は第2のレジスタ群が更新された場合の処理例を示す流れ図、

第4図は第2のレジスタ群が参照された場合の処理例を示す流れ図及び、

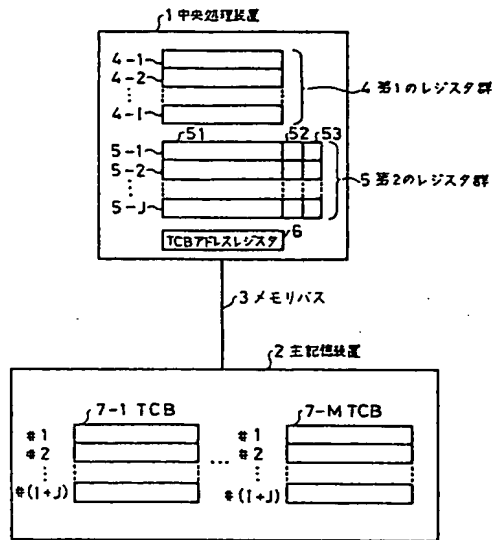
第5図は従来例のブロック図である。

図に於いて、1、10…中央処理装置、2、11…主記憶装置、3、12…メモリアス、4…第1のレジスタ群、5…第2のレジスタ群、4-1～4-I, 5-1～5-J, 13-1～13-N…レジスタ、6、14…TCBアドレスレジスタ、7-1～7-M, 15-1～15-M…TCB、51…環境情報がセットされる領域、52…有効フラグがセットされる領域、53…更新フラグがセットされる領域。

特許出願人 日本電気株式会社

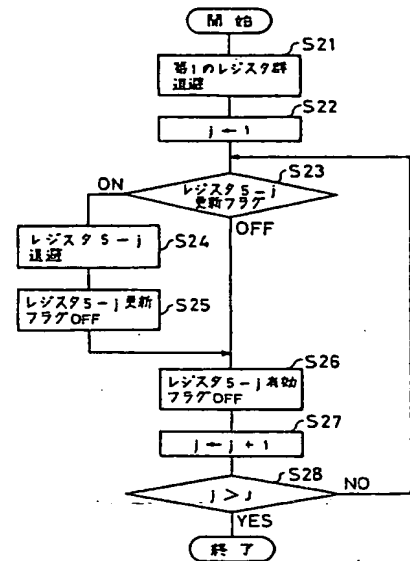
代理人 弁理士 境 廣 巳

特開昭63-314647 (5)



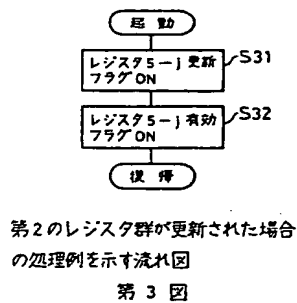
本発明の実施例のブロック図

第1図



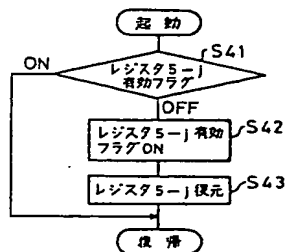
退避時の処理例を示す流れ図

第2図



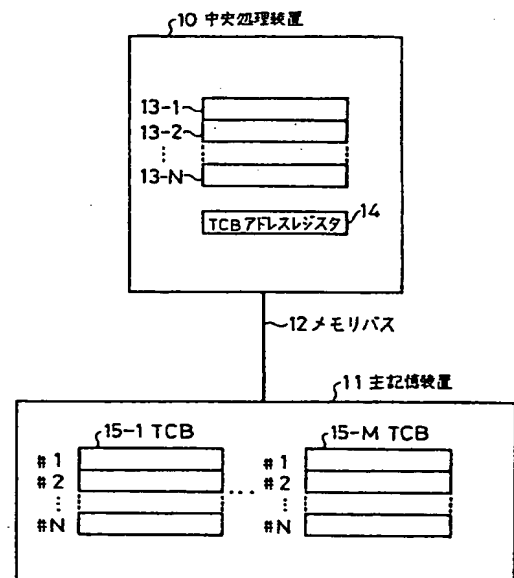
第2のレジスタ群が更新された場合の処理例を示す流れ図

第3図



第2のレジスタ群が参照された場合の処理例を示す流れ図

第4図



従来例のブロック図

第5図